PATENT

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Applicant(s): James J. Babka et al.

Title: EFFICIENT TECHNIQUE FOR MATCHING HIERARCHIES OF ARBITRARY SIZE AND STRUCTURE WITHOUT REGARD TO ORDERING OF ELEMENTS

Application No.: 09/898,708 Filed: July 3, 2001

Examiner: Cam Y T Truong Group Art Unit: 2172

Atty. Docket No.: 027-0006

November 24, 2004

Mail Stop Appeal Brief - Patents
COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, VA 22313

APPELLANT'S BRIEF (37 C.F.R. § 41.37)

This brief is in furtherance of the Notice of Appeal, filed on August 26, 2004. The fee required under § 41.20(b)(2), is provided in the accompanying Transmittal, along with a petition for a one-month extension of time and the fee required under §1.17(a).

REAL PARTY IN INTEREST

The real party in interest in this appeal is Trilogy Development Group, Inc., as evidenced by the assignment recorded at Reel 011970 Frame 0834.

RELATED APPEALS AND INTERFERENCES

Appellants have no knowledge of any related appeals or interferences.

STATUS OF CLAIMS

Claims 1 – 41 are presented herein on appeal. None of the claims have been amended during prosecution. Claims 1 – 41 have been twice rejected in a Non-final Office Action dated

October 9, 2003 and in a Final Office Action dated March 23, 2004. In addition, both Office Actions indicated allowable subject matter in claims 8 – 11 and 16.

Claims 1 – 41, now presented herein on appeal, are reproduced in the Appendix attached hereto.

STATUS OF AMENDMENTS

No amendments after final rejection have been submitted.

SUMMARY OF INVENTION

The invention relates generally to representations of hierarchically organized information and, in particular, to data structures, objects, methods and techniques for matching or, in some realizations, efficiently representing sub-hierarchies thereof. This presently claimed invention relates to an element-order-independent comparison of hierarchically organized data structures that is efficiently performed using a transformation operation that orthogonally and recursively encodes child node information. Orthogonal encodings ensure that a combination (e.g., an additive combination) of values corresponding to elements of a sub-hierarchy is insensitive to ordering of the elements. Recursion can be employed to fold in information contributions at successive layers of an information hierarchy.

Referring to Figures 3 – 5, for example, information for leaf nodes is aggregated recursively with internal nodes at successive levels of a hierarchically organized data structure. These figures depict a basic technique that "includes, for each node in the tree, building a string representation of the node that incorporates information about the node itself plus information about its children, without regard to the order of the children. The basic technique may be employed in the context of tree or sub-tree comparisons, as a precursor to such comparison or as an encoding or transformation technique for tree-oriented data" (paragraph 1021 of Applicant's specification).

In the exemplary illustrations depicted in the figures, "a mapping function is employed to associate each such string representation with a code or key. In general the mapping has attributes that are identical source information maps to identical values and non-identical source

information maps to distinct values. In addition, distinct values so-mapped have the general property that, when considered in the context of a combining operation, they are orthogonally-encoded. Stated differently, the results of such a combining operation performed on first and second sets of values is not the same, unless membership in the two sets is identical, and a result of the combining operation is insensitive to order of operands" (paragraph 1021 of Applicant's specification). For example, referring to Figure 4, "the collapsed form illustrated includes a concatenation of string encoded information associated with the corresponding node (illustrated in FIG. 3) with a string encoding of the arithmetic sum of orthogonally-encoded values for child nodes" (paragraph 1026 of Applicant's specification).

ISSUES

1. Regardless of whether *Jeyaraman*'s disclosed "collapse" operation corresponds to that which Applicant discloses and claims, it is clear that *Jeyaraman*'s disclosed techniques require the use of ordering operations, particularly use of swap operations to order child nodes. Whatever *Jeyaraman*'s "collapse" operation entails, it clearly is not order insensitive with respect to information of the child nodes.

2. *Jeyaraman* does not disclose or suggest orthogonal encodings in any collapsed representations. In effect, the Office simply reads corresponding limitation out of the claims.

3. Properly interpreted, *Jeyaraman*'s collapse operation does not correspond to the scope of the term "collapse" (or "collapsing") as disclosed and claimed by Applicant. *Jeyaraman*'s collapse operation involves an accounting transfer of child nodes appearing in a first sub-tree but not in a second sub-tree into a resulting node that includes the child nodes of both the first and second sub-trees. While any moniker desired can be used to label this transfer mechanism, the mere usage of the same word as Applicant does not mean that *Jeyaraman* discloses or suggests "collapse" (or "collapsing") as disclosed and claimed by Applicant.

4. Properly interpreted, *Jeyaraman* does not disclose or suggest identification of equivalent sub-trees based on correspondence of collapsed representations. Even if

the different use of the term "collapse" is ignored, any matching employed in *Jeyaraman* is of leaf nodes, not collapsed representations.

ARGUMENTS

The rejections of Applicant's claims are supported by mischaracterizations of *Jeyaraman* and disregarding Applicant's claim language. Although U.S. Patent No. 6,311,187 to Jeyaraman is replete with ordering nodes, rejections of Applicant's claims are supported with the mischaracterization that *Jeyaraman* discloses "an unsorted tree; thus, the collapsing the contents of a first node and a second node is in an order-insensitive manner" (page 2, second paragraph of the Final Office Action). Rejections of the claims also suffer from the Office overlooking the actual claims and focusing on a single common word in Applicant's claims and *Jeyaraman*. The "collapsing" disclosed in *Jeyaraman* involves gathering children nodes of separate parent nodes of an old and new tree under a single parent node in the new tree, not collapsing representations of nodes or encodings of nodes as disclosed and claimed by Applicant. The mischaracterization of *Jeyaraman* continues with the simple assumption that *Jeyaraman*'s leaf node matching nodes can support a rejection of Applicant's claims. However, Applicant's claims identify equivalent sub-trees based on collapsed representations. Finally, the Office completely ignores the use of orthogonal encodings in Applicant's claims and supposes the disclosure of such in *Jeyaraman* without actual support for the supposition.

The Office rejects all claims under 35 USC § 103. The Office relies on *Jeyaraman* in rejecting claims 1 – 4, 12, 14, 15, 17, 18, 19, 24, 26 – 32, 35 – 38, and 40 – 41. The Office relies on *Jeyaraman* in view of *Aggarwal* in rejecting claim 5 and relies on *Jeyaraman* in view of *Brown* in rejecting claims 6, 7, 13, 20 – 23, 25, 33, 34, and 39. Specific claimed limitations are not disclosed or suggested in either *Jeyaraman*, *Aggarwal*, or *Brown*. As a result, the prior art has been accorded inordinate scope, the claims have been interpreted in a way that discounts claim limitations, or both. Careful analysis will show that claim limitations are neither expressly nor inherently disclosed in any of the relied upon references. As a result, no *prima facie* case of obviousness has been made out. The nature of this legal error is now summarized.

<u>*Obviousness Rejections under 35 U.S.C. § 103*</u>

Claims 1 – 4, 12, 14, 15, 17, 18, 19, 24, 26 – 32, 35 – 38, and 40 – 41 all stand rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent 6,311,187 to Jeyaraman (*Jeyaraman*). Claims 5 is rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman* in view of U.S. Patent 5,781,906 to Aggarwal et al. (*Aggarwal*). Claims 6, 7, 13, 20 – 23, 25, 33, 34, and 39 are rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman* in view of U.S. Patent No. 6,539,369 to Brown (*Brown*).

The legal standard for obviousness is defined in the Patent Statute, 35 U.S.C. § 103, which specifies, in addition to novelty requirements under § 102, further conditions for patentability relating to nonobvious subject matter. Those further conditions include the following:

> [a] patent may not be obtained though the invention is not identically disclosed or described [by prior art under 35 U.S.C. § 102] if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.

35 U.S.C. § 103 (2004).

Obviousness is a legal determination based on underlying factual inquiries. *Minnesota Min. & Mfg. Co. v. Johnson & Johnson Orthopedics, Inc.*, 976 F.2d 1559, 24 U.S.P.Q.2d 1321, 1332-1333 (Fed. Cir. 1992). *Graham v. John Deere Co.*, 383 U.S. 1, 17 (1966) defines the factual inquiries utilized to evaluate the prior art. Specifically, the prior art is evaluated in terms of: (1) its scope and content; (2) the differences between the prior art and the claimed invention; (3) the level of ordinary skill in the art at the time the application was filed; and (4) objective, or secondary, evidence of nonobviousness such as commercial success, failure of others, long-felt need and unexpected results, which must be considered in reaching a conclusion of obviousness. *Graham v. John Deere Co.*, 383 U.S. 1, 17, 148 U.S.P.Q. 459, 460 (1966); *Panduit Corp. v. Dennison Mfg. Co.*, 810 F.2d 1561, 1566-67, 1 U.S.P.Q.2d 1593, 1595-96 (Fed. Cir. 1987); *Minnesota Min. & Mfg. Co. v. Johnson & Johnson Orthopaedics, Inc.*, 976 F.2d 1559, 24 U.S.P.Q.2d 1321, 1333 (Fed. Cir. 1992). In the present appeal, pertinent issues relate

primarily to specific differences between the prior art and appealed claims, specifically absence, in the relied upon references, of certain features of the appealed claims.

Obviousness analysis begins with a key legal question—what is the invention claimed? In this regard, the claimed invention must be evaluated as a whole. 35 U.S.C. § 103; *see also Panduit Corp.*, 1 U.S.P.Q.2d at 1597, 810 F.2d at 1568. Fundamentally, all claim limitations must be considered in the obviousness analysis. Indeed, it is clear error to ignore limitations clearly set forth in the claims. *Panduit Corp.*, 1 U.S.P.Q.2d, 1603 – 1604, 810 F.2d at 1576. In general, multiple prior art references may be combined to provide a basis for an obviousness determination; however, there must be some teaching or suggestion for the combination. *In re Rouffet*, 149 F.3d 1350, 1355, 47 U.S.P.Q.2d 1453, 1456 (Fed. Cir. 1998). Finally, a prior art reference must be considered in its entirety, *i.e.*, as a *whole*, including portions that would lead away from the claimed invention. *W.L. Gore & Associates, Inc. v. Garlock, Inc.*, 721 F.2d 1540, 1550, 220 U.S.P.Q. 303, 311 (Fed. Cir. 1983). Indeed, it is impermissible within the framework of § 103 to pick and choose from any one reference only so much of it as will support a given position, to the exclusion of other parts necessary to the full appreciation of what such reference fairly suggests to one of ordinary skill in the art. *Bausch & Lomb, Inc. v. Barnes-Hind, Inc.*, 796 F.2d 443, 230 U.S.P.Q. 416 (Fed. Cir. 1986); *see also In re Wright*, 866 F.2d 422, 9 U.S.P.Q.2d 1649, 1652 (Fed. Cir. 1989).

<u>Claims 1, 2, 6, 7, 14 – 15 and 17</u>

Claims 1, 2, 14 – 15, and 17 all stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman*. Claims 6 and 7 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman* in view of *Brown*.

Applicant respectfully submits that the Office ignores the failure of *Jeyaraman* to disclose or suggest collapsing as claimed by Applicant. The Office characterizes *Jeyaraman* as disclosing order insensitive collapsing in contrast to *Jeyaraman*'s actual disclosure. *Jeyaraman* discloses the following:

```
Order the children of 'p' through swap operations. (col.
12, line 6)
```

Applying step 4, the system notices that nodes D0.Se2.P1 and D0.Se0.P1 need to be collapsed. This brings together all the nodes in the SIBLING_LIST under a common parent (See FIG. 6D).
CLP(D0.Se2.P1,D0.Se0.P1)
Next, the system uses swap operation to re-order the nodes (see FIG. 6E). (col. 12, lines 35 - 41)

Use of swap operations to restore order of nodes in the tree is disclosed throughout *Jeyaraman* (col. 13, lines 4 – 9; col. 13, lines 15 – 19; col. 13, lines 25 – 28; col. 14, lines 22 – 25). Even the assignment of node identifiers is order sensitive as seen in Figures 6H – 6J, which depict the node identifiers as exactly reflecting order of the children nodes as they occur in the tree. Hence, it cannot be asserted that *Jeyaraman* discloses "*wherein the collapsing is order-insensitive with respect to information of the respective child nodes*" as recited in claim 1.

In addition, the Office does not address the collapsing as claimed by Applicant. *Jeyaraman* repeatedly discloses ordering nodes in contrast to the interpretation by the Office. The Office states that *Jeyaraman* discloses "an unsorted tree; thus, the collapsing the contents of a first node and a second node is in an order-insensitive manner" (page 2, second paragraph of the Final Office Action) and then again states "this tree is an unsorted tree, thus, the collapsing the contents of a first node and a second node is in an order-insensitive manner" (page 5 of the Final Office Action). To support the mischaracterization of *Jeyaraman*, the Office cites the following sections of *Jeyarman*:

CLP(D0.Se0.P0, D0.Se0.P1). This operation collapses the contents of a first node and a second node. The resulting node gets the same tag types as the first node. The children of the second become the right-most children of the resulting node. (col. 9, lines 20 - 25)

The first parent inherits all of the children that are present in new_t, and the second parent inherits the remaining children. If a parent node in old_t has all of the same children and additional children in new_t, the

> system generates a node collapse operation to bring all the children together in new_t (step 512). (col. 7, lines 61 - 65).
>
> Applying step 4, the system notices that nodes D0.Se2.P1 and D0.Se0.P1 need to be collapsed. This brings together all the nodes in the SIBLING_LIST under a common parent (See FIG. 6D).
> CLP(D0.Se2.P1,D0.Se0.P1)
> Next, the system uses swap operation to re-order the nodes (see FIG. 6E). (col. 12, lines 35 - 41)

The relied upon sections disclose a collapse operation, but not collapsing as claimed by Applicant. The section discloses bringing child nodes together under a single parent node, and does not disclose or suggest "*collapsing plural nodes thereof* ***into respective representations that each incorporate information of a respective node and that of any child nodes thereof***" as recited in claim 1. The Office does not suggest that *Jeyaraman* discloses collapsing in the manner claimed by Applicant, because it is not disclosed by *Jeyaraman*. *Jeyaraman* discloses "assigning a node identity to 'p', which is a collective identity of its latest children" (col. 12, lines 16 – 18), but this assignment is subsequent to the collapse operation and swap operations that restore the tree's order in *Jeyaraman* (col.12, lines 6 – 15).

Furthermore, *Jeyaraman* discloses matching leaf nodes, not matching collapsed representations, in contrast to the Office's arguments. Claim 1 specifically recites "*based on correspondence of particular instances of the collapsed representations, identifying the respective portions as equivalent.*" *Jeyaraman* discloses the following three phases:1) a matching leaf nodes phase, 2) a deletion phase, and 3) a modification phase. The matching phase includes a "first step to generate a unique identifier for each of the leaf nodes in T2 based on the content of the leaf node...Next, the process assigns value identifiers leaf nodes of T1" (col. 9, line 40 – col. 10, line 15). The matching phase disclosed by *Jeyaraman* matches leaf nodes, but does not identify equivalent portions of a tree based on ***collapsed representations.*** Indeed, prior to the modification phase, *Jeyaraman* does not disclose a collapse operation. It is

not until the modification phase that *Jeyaraman* discloses a collapse operation. The modification phase includes indicating collapse operations and assignment of node identifiers after a collapse operation and re-ordering with swap operations (col. 11, line 15 – col. 13, line 51). *Jeyaraman* never discloses or suggests identifying equivalent portions of T1 and T2 with the assigned node identifiers, at least because the matching of equivalent portions has already been performed in the matching of leaf nodes, prior to assignment of node identifiers.

Action Requested: For at least the reasons set forth above, the references fail to teach or suggest, alone or in combination, all elements of claims 1, 2, 6, 7, 14 – 15, and 17. Applicants respectfully request that this Honorable Board reverse the present rejections and direct that the aforementioned claims be issued.

Claims 18, 24, and 26 – 28

Claims 18, 24, and 26 – 28 all stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman.*

Referring to independent claim 18, Applicant respectfully notes that *Jeyaraman* does not disclose or suggest "*wherein the identifiers and combining function are selected to ensure that same combinations of child node identifiers result in same child nodes contributions irrespective of ordering of the child node identifiers*." As already stated, *Jeyaraman* does not disclose identifying nodes irrespective of order. Swap operations are used to restore order of nodes in the tree in the above quoted sections of *Jeyaraman* and throughout *Jeyaraman* (col. 13, lines 4 – 9; col. 13, lines 15 – 19; col. 13, lines 25 – 28; col. 14, lines 22 – 25). Even the assignment of node identifiers is respective of ordering as seen in Figures 6H – 6J, which depict the node identifiers as exactly reflecting order of the children nodes as they occur in the tree.

Action Requested: For at least the reasons set forth above, the relied upon references fail to teach or suggest, alone or in combination, all elements of independent claim 18. Furthemore, all claims 18, 24, and 26 – 28 are allowable and rejections under 35 U.S.C. §103(a) should be reversed. Applicants respectfully request that this Honorable Board reverse the present rejections and direct that the aforementioned claims be issued.

Claims 2 – 5, 13, 19 – 23, 25, 29 – 31

Claims 2 – 5, 19, 29 – 31 all stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman*. Claim 13 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman* in view of *Brown*. Claims 20 – 23 and 25 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman* in view of *Brown*. Claim 5 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman* in view of *Aggarwal*.

As already stated above, the Office relies upon sections of *Jeyaraman* that disclose a collapse operation, but not collapsing as claimed by Applicant. The section discloses bringing child nodes together under a single parent node, and does not disclose or suggest "*recursively collapsing sub-hierarchies thereof using encodings that, at least at a same level thereof, includes orthogonal values*" as recited in claim 29. The Office does not suggest that *Jeyaraman* discloses collapsing in the manner claimed by Applicant, because it is not disclosed by *Jeyaraman*. *Jeyaraman* discloses "assigning a node identity to 'p', which is a collective identity of its latest children" (col. 12, lines 16 – 18), but this assignment is subsequent to the collapse operation and swap operations that restore the tree's order in *Jeyaraman* (col.12, lines 6 – 15). The following sections are relied up by the Office, but do not support the Office's arguments.

To support the mischaracterization of *Jeyaraman*, the Office cites the following sections of *Jeyarman*:

> `CLP(D0.Se0.P0, D0.Se0.P1). This operation collapses the contents of a first node and a second node. The resulting node gets the same tag types as the first node. The children of the second become the right-most children of the resulting node. (col. 9, lines 20 - 25)`

> `The first parent inherits all of the children that are present in new_t, and the second parent inherits the remaining children. If a parent node in old_t has all of the same children and additional children in new_t, the system generates a node collapse operation to bring all`

the children together in new_t (step 512). (col. 7, lines 61 - 65).

Applying step 4, the system notices that nodes D0.Se2.P1 and D0.Se0.P1 need to be collapsed. This brings together all the nodes in the SIBLING_LIST under a common parent (See FIG. 6D). CLP(D0.Se2.P1,D0.Se0.P1)
Next, the system uses swap operation to re-order the nodes (see FIG. 6E). (col. 12, lines 35 - 41)

Furthermore, the Office completely disregards the limitation of ***orthogonal encodings*** and presumes its disclosure. Nothing in *Jeyaraman* can be found that discloses or suggests orthogonal encodings, much less orthogonal encodings as claimed by Applicant. The Office cites the following section of *Jeyaraman*.

The modification phase brings together the children of internal nodes, in a bottom-up-fashion. This involves scanning all the nodes from the bottom-most level (furthest from the root), and scanning each level until level zero is reached. Note that the identity of each internal node is established by the collective identity of its children. For example, if a parent node's children are identifies as 'a' and 'b' respectively, then the identity of the parent is 'ab.'

Also, if a parent node is left with no children as a result of a move operation, the parent node is deleted. Furthermore, in the special case where there is a skewed tree r sub-tree of nodes having just one child, i.e., a→b→c→d, when node 'd' is deleted, node 'c' is also be deleted. This action is repeated until node ;a; is deleted as well, Instead of generating an individual delete operation for each one of the nodes, the chain of delete operations is reduced to a single delete operation of the grandest common parent of all nodes being deleted.

```
Pseudo-code for one embodiment of the modification phase appears
below.
For each level_i, in T2 (leaf to the root) {
1. TO_BE_COMPLETED_LIST = list of all the node value identifiers
at level_i in T2.
2. If the node in the TO_BE_COMPLETED_LIST is the root node, find
the matching node 't' in T1'. If 't' happens is root node, break
from the loop.
Else, partition T1' into two nodes, such that the sub-tree rooted
at 't' is moved away from T1' and becomes another tree (T1'').
Next, delete the source partition (T1') by deleting its grandest
common parent (the root). T1'' and T2 are now identical.
```

Nothing in the entire section relied upon by the Office makes reference to orthogonal encodings. *Jeyaraman* does not disclose or suggest orthogonal encoding and especially does not disclose or suggest collapsing using the encodings of the nodes that include orthogonal values.

The Office simply refers to *Jeyaraman*'s collapse operation, but *Jeyaraman*'s collapse operation does not collapse using encodings that include orthogonal values. Essentially, the Office simply reads out limitations corresponding to orthogonal encodings out of the claims.

Action Requested: For at least the reasons set forth above, the relied upon references fail to teach or suggest, alone or in combination, all elements of independent claim 29. In addition, all claims 2 – 5, 13, 19 – 23, 25, 29 – 31 are allowable and rejections under 35 U.S.C. §103(a) should be reversed. Applicants respectfully request that this Honorable Board reverse the present rejections and direct that the aforementioned claims be issued.

Claims 33 – 35 and 38 – 39

Claims 33 – 34 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman* in view of *Brown*. Claims 35 and 38 – 39 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman*.

As stated above, *Jeyaraman* does not disclose or suggest order-insensitive collapsing. *Jeyaraman* specifically and repeatedly describes ordering trees. Further, despite the use of the

term "collapse," *Jeyaraman* describes a collapse operation that shuffles around leaf nodes, but does not disclose or suggest aggregating contents of leaf nodes. Referring to independent claims 33 and 35, Applicant respectfully submits that *Jeyaraman* does not disclose or suggest "*instructions, when executed, causing the processor to define a counterpart data structure by collapsing plural nodes of the hierarchically-organized data structure into respective representations that each incorporate information of a respective node and that of any child nodes thereof,* ***wherein the collapsing includes an order-insensitive aggregation of orthogonal encodings of information of the respective child nodes***." As already stated, *Jeyaraman* does not disclose or suggest collapsing that includes aggregation of child node information and does not disclose or suggest order-insensitive aggregation of orthogonal encodings of information of child nodes.

Action Requested: For at least the reasons set forth above, the relied upon references fail to teach or suggest, alone or in combination, all elements of independent claim 33 and 35. Accordingly, all claims 33 – 35, and 38 – 39 are allowable and rejections under 35 U.S.C. §103(a) should be reversed. Applicants respectfully request that this Honorable Board reverse the present rejections and direct that the aforementioned claims be issued.

Claim 41

Claim 41 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman*.

As already stated, Jeyaraman does not disclose or suggest order-independent comparisons or orthogonal encodings. Jeyaraman specifically and repeatedly discloses employment of operations to order nodes. Referring to independent claim 41, Applicant respectfully submits that *Jeyaraman* does not disclose or suggest "*means for performing an* ***element order independent comparison of hierarchically organized data structures*** *using a transformation operation that orthogonally and recursively encodes child node information*." The Office argues the following:

> However, Jeyaraman teaches that as mapping leaf nodes tree 1 and tree 2. For each level_I in T2 (leaf to the root) {to_be_completed_list=list of all the node value

identifiers at level_I in T2. If the node in the to_be_completed_list is the root node, find the matching node t in T1. Where the leaf nodes actually contain data and value identifiers. The modification phase brings together the children of internal nodes, in a bottom-up fashion. This involves scanning all the nodes from the bottom-most level and scanning each level until level zero is reached. Note that the identity of each internal node is established by the collective identify of its children. The system transforms old_tree into new_tree. The above information shows the system maps the contents of nodes in T1 and T1 recursively by following a bottom up fashion.

It would have been obvious to a person of an ordinary skill in the art at the time the invention was made to apply Jeyaraman's teaching of scanning all the nodes from bottom-most level to compare leaf nodes in tree for transforms old_t into new)t in order to update a tree structure.

Applicant notes that the entire rational for rejecting claim 41 is devoid of any discussion related to element order independent comparison. The Office addresses recursion and scanning, but never addresses element order independent comparison. As already stated, *Jeyaraman* does not disclose or suggest order independence.

Action Requested: For at least the reasons set forth above, the relied upon references fail to teach or suggest, alone or in combination, all elements of independent claim 41. Accordingly, claim 41 is allowable and rejections under 35 U.S.C. §103(a) should be reversed. Applicants respectfully request that this Honorable Board reverse the present rejections and direct that the aforementioned claims be issued.

Claims 3 – 4

Claims 3 – 4 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman*.

Referring to claims 3 and 4, Applicant respectfully submits that *Jeyaraman* does not disclose or suggest "*wherein a unit of orthogonally-encoded child node information* ***includes a power-of-two encoded mapping*** *of a concatenation of the child node information*." The Office states that "data and value identifier is presented as a power-of-two encoded value" to support rejection of claims 3 and 4. For support, the Office refers to the following sections of *Jeyaraman*:

> There are three phases in the process, including: (1) matching the leaf nodes in T1 and T2; (2) deleting nodes in T1 with no match in T2; and (3) modifying or moving nodes the remaining nodes to create T1.
>
> Phase 1: Matching Leaf Nodes
> The first step is to generate a unique identifier for each of the leaf nodes in T2 based on the content of the leaf node. This can be accomplished by using a hash function to generate a unique identifier for each of the leaf nodes. If two leaf nodes have the same content, then the hash function generates the same identifier. (col. 9, lines 35 - 46)
>
> All nodes in FIG. 6A include a name (tag, a value, and an associated value identifier. Since the leaf nodes actually contain data, value identifiers are assigned to them before the process starts; whereas, for an internal node, a value identifier is assigned during the comparison process based upon the value of identifiers of the internal node's children. (col. 8, lines 20 - 25)

Neither the relied upon section nor any other section of *Jeyaraman* disclose or suggest the subject matter of claims 3 and 4. The Office's only support for rejection of the claims is citation to sections that disclose no more than assigning value identifiers using a hash function and never disclose or suggest power-of-two encoding.

Action Requested: For at least the reasons set forth above, the relied upon references fail to teach or suggest, alone or in combination, all elements of claims 3 and 4. Accordingly, claims 3 and 4 are allowable and rejections under 35 U.S.C. §103(a) should be reversed. Applicants respectfully request that this Honorable Board reverse the present rejections and direct that the aforementioned claims be issued.

Claim 5

Claim 5 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman* in view of *Aggarwal.*

Referring to claim 5, Application respectfully submits that neither *Jeyaraman* nor *Aggarwal*, standing alone or in combination, discloses or suggest "*wherein the order-insensitive collapsing* ***includes an arithmetic sum of orthogonal binary encodings*** *of child node information*." The Office asserts that

> [i]t would have been obvious to one skilled in the art at the time the invention was made to apply Aggrawal's (sic) teaching of in a binary tree, the leaf numbers N.sub.B.sbsb.1 and N.sub.B.sbsb.2 of each of these children are eat at least p.multidot.N.sub.A. Among all partitions examined which satisfy this leaf number condition, the one chosen minimizes the sum of the areas of the minimum bounding rectangles A(I.sub.B.sbsb.I) + A(I.sub.B.sbsb.I) subject to Jeyaraman's system in order to save memory space.

Applicant respectfully submits that the Office has misapplied and mischaracterized *Aggarwal*. The binarization of a tree disclosed by *Aggarwal* relates to construction of a binary tree in a top-down fashion "such that the entries in the leaf nodes correspond to multi-

dimensional (also called spatial) data objects stored on DASD 105" (col. 5 lines 51 – 54). The binarization disclosed by *Aggarwal* relates to the structure of a constructed binary tree and not to the contents of nodes or encodings of child node information.

Action Requested: For at least the reasons set forth above, the relied upon references fail to teach or suggest, alone or in combination, all elements of claim 5. Accordingly, claim 5 is allowable and rejections under 35 U.S.C. §103(a) should be reversed. Applicants respectfully request that this Honorable Board reverse the present rejections and direct that the aforementioned claims be issued.

Claims 36, 37, 40

Claims 36, 37, and 40 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over *Jeyaraman*.

As previously stated, *Jeyaraman* discloses

> three phases in the process, including: (1) matching the leaf nodes in T1 and T2; (2) deleting nodes in T1 with no match in T2; and (3) modifying or moving nodes the remaining nodes to create T1.
>
> Phase 1: Matching Leaf Nodes
> The first step is to generate a unique identifier for each of the leaf nodes in T2 based on the content of the leaf node. This can be accomplished by using a hash function to generate a unique identifier for each of the leaf nodes. If two leaf nodes have the same content, then the hash function generates the same identifier. (col. 9, lines 35 - 46)
>
> All nodes in FIG. 6A include a name (tag, a value, and an associated value identifier. Since the leaf nodes actually contain data, value identifiers are assigned to them before the process starts; whereas, for an internal

node, a value identifier is assigned during the comparison process based upon the value of identifiers of the internal node's children. (col. 8, lines 20 - 25)

The matching phase disclosed by *Jeyaraman* does not identify equivalent portions, distinct sub-hierarchies, or equivalent sub-assemblies based on collapsed representations of respective nodes that each incorporate information of a respective node and that of any child nodes thereof. Indeed, prior to the modification phase, Jeyaraman does not disclose a collapse operation. It is not until the modification phase that *Jeyaraman* discloses a collapse operation. The modification phase includes indicating collapse operations and assignment of node identifiers after a collapse operation and re-ordering with swap operations (col. 11, line 15 – col. 13, line 51). *Jeyaraman* never discloses or suggests

> *matching instructions executable by the one or more processors* ***to identify distinct sub-hierarchies of the hierarchically-organized data structure as at least equivalent based on correspondence of the collapsed representations****; (claim 36)*
>
> *matching instructions executable by the one or more processors* ***to identify at least equivalent portions of first and second ones of the hierarchically-organized data structure based on correspondence of collapsed representations thereof****; (claim 37) and*
>
> *wherein the information management tool further* ***identifies, based on correspondence of collapsed representations of the hierarchically-organized data structure, equivalent sub-assemblies without regard to ordering of elements thereof*** *(claim 40).*

Action Requested: For at least the reasons set forth above, the relied upon references fail to teach or suggest, alone or in combination, all elements of claims 36, 37, and 40. Accordingly, claims 36, 37 and 40 are allowable and rejections under 35 U.S.C. §103(a) should be reversed. Applicants respectfully request that this Honorable Board reverse the present rejections and direct that the aforementioned claims be issued.

CONCLUSIONS

For at least the foregoing reasons, Appellants' claimed invention would not have been obvious under 35 U.S.C. §103(a) over the cited prior art. Accordingly, this Honorable Board is respectfully requested to reverse the rejection of claims 1 – 41 and direct this application to be issued.

CERTIFICATE OF MAILING OR TRANSMISSION

I hereby certify that, on the date shown below, this correspondence is being

☐ deposited with the US Postal Service with sufficient postage as first class mail, in an envelope addressed to Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

☒ facsimile transmitted to the US Patent and Trademark Office.

Nov-24-2004
Date

EXPRESS MAIL LABEL: ______________

Respectfully submitted,

Steven R. Gilliam, Reg. No. 51,734
Attorney for Applicant(s)
(512) 338-6320
(512) 338-6301 (fax)

APPENDIX OF CLAIMS INVOLVED IN THE APPEAL

1. (Original) A method of identifying equivalent portions of one or more unsorted hierarchically-organized data structures, the method comprising:

collapsing plural nodes thereof into respective representations that each incorporate information of a respective node and that of any child nodes thereof; and

based on correspondence of particular instances of the collapsed representations, identifying the respective portions as equivalent,

wherein the collapsing is order-insensitive with respect to information of the respective child nodes.

2. (Original) A method as recited in claim 1,

wherein the collapsed representations include respective aggregations of orthogonally-encoded child node information.

3. (Original) A method as recited in claim 2,

wherein a unit of orthogonally-encoded child node information includes a power-of-two encoded mapping of a concatenation of the child node information with a similarly encoded mapping of respective information of child nodes thereof.

4. (Original) A method as recited in claim 2,

wherein a unit of orthogonally-encoded child node information includes a power-of-two encoded mapping of a concatenation of the child node information with recursively encoded mappings of respective sub-hierarchies thereof.

5. (Original) A method as recited in claim 1,

wherein the order-insensitive collapsing includes an arithmetic sum of orthogonal binary encodings of child node information.

6. (Original) A method as recited in claim 1,

wherein distinct tables are defined for each level of the hierarchically-organized data structure.

7. (Original) A method as recited in claim 1,
wherein a table spans multiple levels of the hierarchically-organized data structure.

8. (Original) A method as recited in claim 1, wherein, at a particular node of the hierarchically-organized data structure, the order-insensitive collapsing includes:
an arithmetic addition of orthogonal binary encodings that identify corresponding table entries for respective child nodes; and
concatenation of a result of the arithmetic addition with an encoding of information for the particular node.

9. (Original) A method as recited in claim 8,
wherein the order-insensitive collapsing at the particular node further includes creating a new mapping of the concatenation, the new mapping being an encoding that is at least orthogonal with that for any other node at a same level of the hierarchically-organized data structure.

10. (Original) A method as recited in claim 8,
wherein the order-insensitive collapsing at the particular node further includes creating a new mapping of the concatenation, the new mapping being an encoding that is orthogonal with that for any other node of the hierarchically-organized data structure.

11. (Original) A method as recited in claim 8,
wherein the encoding of particular node information is a string encoding thereof.

12. A method as recited in claim 1,
wherein the correspondence collapsed representations is based on identity of respective mapped codes.

13. (Original) A method as recited in claim 1,
wherein the order-insensitive collapsing includes an arithmetic addition of orthogonally-encoded values that index into a store of child node information.

14. (Original) A method as recited in claim 1,
wherein the hierarchically-organized data structure includes at least three levels of nodes; and
further comprising performing the collapsing at successive ones of the levels of the hierarchically-organized data structure.

15. (Original) A method as recited in claim 1,
wherein the hierarchically-organized data structure includes a tree-organized data structure.

16. (Original) A method as recited in claim 8,
wherein the hierarchically-organized data structure includes at least two levels.

17. (Original) A method as recited in claim 1,
wherein the hierarchically-organized data structure encodes subassembly information as sub-hierarchies thereof and encodes component parts at least at leaf nodes thereof.

18. (Original) A method of identifying equivalent logical sub-trees of a tree-oriented data representation, the method comprising:
associating a first-level identifier with each of plural leaf nodes at a first-level of the tree, wherein distinct leaf node values are associated with distinct first identifiers and equivalent leaf node values are associated with same first identifiers; and
at each next level of the tree, associating an identifier with each node thereof, each such identifier including a current node contribution and a contribution associated with any child nodes thereof,
wherein the child nodes contribution is computed using a combining function operative on identifiers associated with the child nodes,

wherein the identifiers and combining function are selected to ensure that same combinations of child node identifiers result in same child nodes contributions irrespective of ordering of the child node identifiers, and

wherein for a second level of the tree, respective child nodes are the leaf nodes of the first-level of the tree.

19. (Original) A method as recited in claim 18,

wherein the identifiers are orthogonally-encoded mappings of respective string encodings of the current node contribution concatenated with respective orthogonally-encoded mappings of child node information.

20. (Original) A method as recited in claim 18,

wherein the orthogonally-encoded mappings at each level of the tree-oriented data representation are in accordance with a corresponding level-specific table.

21. (Original) A method as recited in claim 18,

wherein the orthogonally-encoded mappings for distinct portions of the tree-oriented data representation are in accordance with respective tables.

22. (Original) A method as recited in claim 18,

wherein the orthogonally-encoded mappings for multiple levels of the tree-oriented data representation are in accordance with a single corresponding hash table.

23. (Original) A method as recited in claim 18,

wherein the orthogonally-encoded hashes for each level of the tree-oriented data representation are in accordance with a single corresponding table.

24. (Original) A method as recited in claim 18,

wherein, at least at any particular level of the tree-oriented data representation, the identifiers are orthogonally-encoded.

25. (Original) A method as recited in claim 18,

wherein the identifiers correspond to orthogonal binary encodings of integers; and

wherein the combining function includes addition.

26. (Original) The method of claim 18, employed in a duplicate elimination operation on the tree-oriented data representation.

27. (Original) The method of claim 18, employed in a duplicate identification operation on the tree-oriented data representation.

28. (Original) The method of claim 18, employed in an equality test operation on portions of the tree-oriented data representation.

29. (Original) A method of representing hierarchically-organized data, the method comprising:

recursively collapsing sub-hierarchies thereof using encodings that, at least at a same level thereof, includes orthogonal values;

representing any given node of the hierarchically-organized data as a concatenation of node-specific information with a combination of the orthogonal values for each collapsed sub-hierarchy therebeneath.

30. (Original) The method of claim 29,

transforming from a first encoding of the hierarchically-organized data to a collapsed second form.

31. (Original) The method of claim 29,

employed to eliminate duplicate sub-hierarchies in the hierarchically-organized data.

32. (Original) The method of claim 29,

employed to collapse duplicate sub-hierarchies in the hierarchically-organized data, wherein the concatenation further includes a count of duplicate sub-hierarchies collapsed beneath any given node.

33. (Original) A computer program product encoded in at least one computer readable medium, the computer program product comprising:

a program sequence including a recursively called set of instructions executable by one or more processors to operate on at least one instance of an hierarchically-organized data structure, the instructions, when executed, causing the processor to define a counterpart data structure by collapsing plural nodes of the hierarchically-organized data structure into respective representations that each incorporate information of a respective node and that of any child nodes thereof, wherein the collapsing includes an order-insensitive aggregation of orthogonal encodings of information of the respective child nodes; and

an object implementing the counterpart data structure including at least one table wherein values thereof provide the orthogonal encodings and keys thereof combine the information of respective nodes with an aggregation of the collapsed representations for child nodes thereof.

34. (Original) The computer program product of claim 33,

wherein the at least one computer readable medium is selected from the set of a disk, tape or other magnetic, optical, or electronic storage medium and a network, wireline, wireless or other communications medium.

35. (Original) An information management tool including software executable by one or more processors, the information management tool comprising:

an encoding of a hierarchically-organized data structure instantiable in memory addressable by the one or more processors;

instructions executable by the one or more processors to operate on at least one instance of the hierarchically-organized data structure instantiated in memory, the instructions, when executed, causing the processor to define a counterpart data structure in the memory by collapsing plural nodes of the hierarchically-organized data structure into respective representations that each incorporate information of a respective node and that of any child nodes thereof, wherein the collapsing

includes an order-insensitive aggregation of orthogonal encodings of information of the respective child nodes.

36. (Original) An information management tool, as recited in claim 35, further comprising:

matching instructions executable by the one or more processors to identify distinct sub-hierarchies of the hierarchically-organized data structure as at least equivalent based on correspondence of the collapsed representations.

37. (Original) An information management tool, as recited in claim 35, further comprising:

matching instructions executable by the one or more processors to identify at least equivalent portions of first and second ones of the hierarchically-organized data structure based on correspondence of collapsed representations thereof.

38. (Original) An information management tool, as recited in claim 35,

wherein the order insensitive aggregation is performed recursively at successive levels of a collapsed sub-hierarchy.

39. (Original) An information management tool, as recited in claim 35, wherein the counterpart data structure includes:

at least one hash table; and

a recursively encoded mapping wherein, for any particular node of the hierarchically-organized data structure, a corresponding table entry encodes both respective values for child nodes thereof in accordance with the order-insensitive information and aggregation associated with the particular node itself, and wherein, at least for same-level nodes of the hierarchically-organized data structure, corresponding values are orthogonal.

40. (Original) An information management tool, as recited in claim 35,

wherein the hierarchically-organized data structure encodes and a sub-assembly decomposition of a product configuration; and

wherein the information management tool further identifies, based on correspondence of collapsed representations of the hierarchically-organized data structure, equivalent sub-assemblies without regard to ordering of elements thereof.

41. (Original) An apparatus comprising:

a processor and memory addressable thereby; and

means for performing an element order independent comparison of hierarchically organized data structures using a transformation operation that orthogonally and recursively encodes child node information.